SAM data handling certification and testing

The purpose of this document is to define the procedure to test and certify future SAM data handling service releases. In this document I will list basic data handling use cases employed at CDF and D0, define critical certification parameters as well as set the appropriate measures of the certification success.

Certification

The purpose of the certification testing is to match the system releases against specific set of critical requirements dictated by the experiment. The certification tests must be able to model in real time system use as well as define and quantify all input and output parameters that are important for the particular test/use case. In this document such parameters are referred to as certification parameters.

Certification steps

We suggest formalizing the certification procedure following way:

1. For each deployment pattern we define set of input certification parameters and their values
2. The set is then used to drive “test harness” framework in order to exercise relevant functions of the SAM release.
3. Test harness framework records and quantifies the system output.
4. The system output is numerically compared against “etalon”. “Etalon” must be generated in a similar environment with one of the older, certified releases.
5. The deviation is studied and compared to limits acceptable by the current deployment pattern.

Common certification parameters

Certification parameters attempt to quantify major or most commonly met system events and environmental factors that define and describe SAM data handling performance and its quality of service. For simplicity, we divide certification parameter types into rates and absolute values.

Rate parameters describe probability of station related events within certain time interval. For simplicity, we assume that the event occurrence follows a Poisson process.
The absolute values describe the magnitude of common system variables that influence station interface response to related events. Absolute values are random and in our model follow normal distribution with pre-set mean and standard deviation.
Below is the proposed list of the input and output certification parameters:

1. Input parameters
    a. Number of projects per day.
    b. Number of files per project
    c. Number of consumer processes per project,
    d. Number of station restarts per day
    e. Number of DB server failures per day
    f. Time to transfer a file
    g. Transfer error rate
    h. Project file overlap
    i. Time to release a file
2. Output parameters
    a. Data to consumer delivery rate.
    b. Consumer error rate
    c. Project completion rate.
    d. Number of new core files.

Measure of success

At the moment, we don't have a good say on the expectations of system performance in any of the following setups. Therefore, we plan to employ "staged" approach based on existing stable release to establish "workable" certification input and output etalons for all covered deployment patterns.
Clearly, lack of resources will not allow us to reproduce the production in a simulated environment verbatim. That is why; the first stage of the process is to scale the input from the running production of the D0 and CDF in order to fit the test setups into resources at

hand, yet preserving appropriateness. Several iterations may be required to achieve sustainable and reproducible test setup.

Test harness

The goal of the test harness framework is to accept formalized input based on values of the certification parameters, drive the SAM data handling by generating relevant events and simulating environmental factors in real time, store and analyze the system output and present the “naked eye” outcome of the test.

Events

Start project

sam start project –defname=<definition name > --group=<group> --project=<project name>

Notes on input arguments:

Definition name should be generated dynamically according to the “overlap” certification parameter

Create consumer

sam create consumer --appName=<value> --appVersion=<value> --project=<project name> --station=<value>

output value : consumer id

Create consumer process

sam establish consumer process --consumerId=<consumer id> --project=<value> --station=<value> --limit=<max number of files to deliver>

output value: process id

Create persistent consumer process

sam establish persistent consumer process --consumerId=<consumer id> --outputfile=<value> --project=<value> --station=<value> --outputfile=<file name or file descr id> --node=<node> --limit=<max number of files to deliver>

output value : process id a

Get next file

sam get next file --processId=<process id> --project=<value> --station=<value>
output value : file path

Release next file

sam release file --fileName=<file path> --processId=<process id> --project=<value> --station=<value> --status=<value>

Transfer file

An important aspect of the test harness framework is its ability to reproduce the results of a certain test case within acceptable deviation limits. In order to achieve such reproducibility, the test harness framework must be able control occurrence and probabilities of events that influence the running system. An aspect of that is file transfer events.
File transfer is the external call out triggered by the station. This outcome of the callout is controlled by many hard to reproduce environmental factors such as: disk space availability, network state, machine load. To mitigate uncertainty that apply to the above factors, we suggest employing a file transfer simulator tool. The tool should give better control over occurrence and probability of factors that influence the outcome of the data transfers.
The tool must be implemented as samcp module to create 0 content files of the specified size instead of attempting to perform de-facto transfer.
CRC checking must be disabled in order to be able to use 0 content files in place of the real data.

Fail to transfer file

The dummy transfer tool should also be able to simulate several failure modes:

1. Transient failure
2. No Space failure
3. Timeout failure

DB server failure

Terminate DB server with -9 signal

Station restart

Terminate station with -9 signal

CDF

Test setup

CDF CAF setup is the most common setting for the CDF analysis. For certification and testing purpose we will attempt to replicate and scale the CDF CAF environment into one of the usecases supported by the test harness.
https://plone3.fnal.gov/SAMGrid/Wiki/SAMOnCAFSetup

Certification parameters

TBD

Acceptable deviation

TBD

D0

Test setup

Clearly, with more than 300 nodes in operation, CAB environment is impossible to replicate on the resources at hand. The presented approach attempts to virtualize the setup by defining dummy nodes operated by designated SAM stagers with help of samcp simulators. The virtualization stresses the system interface implementations in a minimalist environment.
SAM station portion of the setup is here:
https://plone3.fnal.gov/SAMGrid/Wiki/SAMOnCabSetup

Certification parameters

Acceptable deviation

Offsite, optional

Basic features.,